Woman C.P.A.

Volume 53 | Issue 1 | Article 6

Winter 1991

EDP Department: Microcomputers in Operating Departments: Controlling the Risks

William Paxton

Elise Jancura

Recommended Citation

Paxton, William and Jancura, Elise (1991) "EDP Department: Microcomputers in Operating Departments: Controlling the Risks," *Woman C.P.A.*: Vol. 53 : Iss. 1 , Article 6.
Available at: https://egrove.olemiss.edu/wcpa/vol53/iss1/6

EDP Department

Elise Jancura, Editor
Cleveland State University

Microcomputers in Operating Departments: Controlling the Risks

By William Paxton, DBA, CPA

Control procedures such as documentation, program and data entry validation, and backup procedures followed by MIS, EDP and accounting departments provide a high degree of reliability for data originating from these sources. The proliferation of microcomputers (PC's) outside traditional data processing departments has resulted in operating managers using data for economically important decisions that have not been subject to such controls and, therefore, may be of questionable quality.

A manager's decisions will be no better than the data on which they are based. "Managers find the personal computer to be a strong yet flexible aid in analyzing complex marketing, financial and manufacturing data ... however, if incorrectly used, it could cost more money than it saves" [Merino, 1983]. Managers have come to rely on the quality of EDP output to the point that reports printed by computers are generally assumed to be accurate. This trust may be misplaced in the case of the output of PC's located outside the control of data processing and accounting departments.

Examples of spreadsheet disasters are not hard to find, and some have been widely publicized in the business press [Kseniak, 1984] [PC Week, 1986] [Howitt, 1985]. A few examples are presented below to illustrate the possible damage when simple checks and controls are omitted.

1. Executives from a Dallas-based oil and gas company were fired because errors in a spreadsheet cost the company several million dollars during a major acquisition.
2. An inventory manager used outdated data and ordered 30,000 parts at $4 each when current requirements were for only 1,500 parts.
3. A division of a large manufacturing firm had its payroll on a spreadsheet that had worked well for nearly a year. Minor modifications were made and the spreadsheet was only partially checked because of its history of reliable operation. A few months later it was discovered that the revised spreadsheet had given employees an extra nickle an hour raise above what they were entitled to, costing the firm over $10,000.
4. A small company used Lotus spreadsheets for its accounting system. It had to go back to a manual system when the only person who knew how to run the package quit. The company nearly went out of business while it was setting up a replacement manual system.
5.A firm had a critical spreadsheet application that failed after some needed modifications were made. The person who had created the spreadsheet had left the firm and there was little documentation of the program. The firm had to engage a consultant to get the application running again. The cost was high and urgent work was delayed.
6. A firm had its mailing list on a spreadsheet. Its vulnera-

bility was exposed when it tried to do a mailing while the person who normally ran the application was on vacation. They couldn't even find the template disk, let alone run the program.

Each of these examples presents errors which may be more costly to firms than intentional misuse of microcomputers.

Extent of the Problem

Published estimates that 30% or more of spreadsheets have errors [Creeth, 1985] [Greenberg, 1986] [Howitt, 1985] indicate that these are not isolated instances. It is obvious from the cost of the errors that this is a problem that should not be ignored. Good control procedures are needed to bring critical PC output to the point where it deserves the trust placed in traditional computer generated data.

It would seem that the extent of errors in PC generated output and the potential damage they can cause would motivate firms to act decisively to control these risks. This has not yet happened. Over seventy percent of the respondents in a recent survey of large publicly held firms indicated they had no controls over the development and use of microcomputer-based programs. Less than four percent indicated microcomputer applications were reviewed by an internal audit group prior to use. The other controls mentioned were essentially controls on the cost of developing an application, not quality control [Powell and Strickland, 1989].

It would seem that the extent of errors in PC generated output and the potential damage they can cause would motivate firms to act decisively to control these risks. This has not yet happened.

Every spreadsheet or data base program (application) doesn't warrant submission to a full set of formal procedures before its use. Control procedures should be limited to applications where there is a favorable cost/benefit relationship. These would frequently include programs that affect recording of assets, ordering, initiation of payments, or that generate data whose uses may not be fully known. Programs whose output will be used frequently or whose output affects major decisions are also likely to satisfy cost/benefit requirements.

Adaptations of verification, documentation and backup techniques well known to data processing professionals would prevent most PC disasters just as they have prevented most of the potential disasters in conventional computing. These techniques generally are not applied to PC applications outside the accounting and data processing departments because they are not known to most persons whose expertise lies in other functional areas such as production or marketing. There also is a natural tendency for people to look toward the next task rather than check and document their work. Therefore, there is a need to establish standards and control procedures for critical PC applications.

The next section presents a documentation based quality control program to help reduce the risks associated with using data from microcomputers. The third section discusses the three ways managers are vulnerable when relying on microcomputer generated data and shows how the quality control program protects against these areas of vulnerability. Finally, the control process and its advantages are summarized.

Spreadsheet applications are generally used for illustration purposes throughout this paper, but similar considerations apply to other types of PC applications such as data base management and financial planning packages.

II. Control Procedures for Critical PC Applications

The control procedures outlined in this paper should be applied to those programs that can have a material effect on the economic performance or financial reporting of the firm. Just as purchasing departments implement different procedures depending on the nature, frequency and size of purchases, QCP administrators will adopt different procedures for spreadsheets and other programs depending on factors such as the frequency of use and magnitude of the impact an error can have on the firm. The set of procedures appropriate to individual firms can be developed internally or by consultants such as CPA's. The procedures should be reviewed and updated periodically.

The Accountant's Role in the QCP

Although PC based applications are a significant and growing part of corporations' information resources, they are often outside the formal information system of the firm. It is necessary to bring the most critical of these programs into the formal information system network of the firm. Otherwise, internal control is compromised and there is no way to assure the quality of the programs being used to support critical decisions.

AIS should establish criteria determining which programs are to be subject to the control procedure.

Accountants are familiar with information requirements of businesses, microcomputer programs, internal control, and testing of computerized applications. This makes the accounting information systems (AIS) function a natural candidate to administer the QCP.

AIS should establish criteria determining which programs are to be subject to the control procedure. Firm wise standards should be established for program validation, documentation and control procedures. AIS should evaluate validation tests and documentation submitted for critical programs, maintain backup copies of approved programs and related documentation, and issue a directory of tested applications with their identification codes.

The internal audit department should include monitoring compliance with policy as part of its normal evaluation of internal control. Including evaluation of the microcomputer internal control process in the audit program will send a signal to operating departments indicating the importance of the process to top management and the firm.

Environmental Constraints

Control procedures must take into consideration environmental factors surrounding the use of PC's. Factors driving the increased use of PC's include:

- the ability to get programs up more rapidly than by going through the EDP department
- flexibility of PC software
- ability to modify programs quickly and easily
- costs that are lower than charges from an EDP department. A control program that significantly reduces these advantages is likely to be circumvented.

The QCP minimizes conflict with users by limiting the program to critical applications. In these applications the QCP operates primarily by requiring documentation that will be readily available if good program development procedures are being followed. The incentive for managers to insist on qualified programs is provided by holding them responsible for the effects of errors attributable to their use of non-qualified data.

Two points need to be noted here. First, the QCP is not directed at fraud. There are techniques to deal with such problems, but they are beyond the scope of this paper. Second, the QCP is not intended to prevent managers from using data produced by non-approved programs. The QCP allows managers to identify data as coming from a qualified program or not. They can then adjust their decision process to take the appropriate risk into consideration.

The QCP from the perspective of the data user

Data users are provided with a list of qualified programs and their identification codes. Each qualified program includes its code number as part of its output. The user merely needs to check the code, if any, on printouts he/she intends to use against the public listing of program identification codes. If the codes match, the user knows that the program generating the data has met QCP standards.

Elements of the QCP

A typical QCP will include program validation, evaluation of program, operator and data user documentation and program design criteria. The QCP administrator will maintain copies of program disks, program validation tests, and program, operator and user documentation. He/she will also issue identification codes to qualifying programs and distribute the identification codes of qualifying programs to potential users.

Program Documentation

Program documentation will generally involve the following elements:

- Definition of variables used in the program
- an overview of program operation including major assumptions and limitations
- an explanation of the operation of each block of program code where a block could be a single complicated macro
- a list of required input data and the source of these inputs
- a description of the program's output.

Program documentation is a key element in implementing control for erroneous data. Individual macros should be thoroughly described as to both function and detailed operation. Programs that are not well documented are very difficult to test and change even if the original author is still available to the firm, and next to impossible to deal with if he/she is not available.

It may be more costly to debug or update a poorly documented program than to generate a new program from scratch. Program maintenance over the life of a program can amount to several times the cost of writing the original program. Good documentation and program design can help cut these costs substantially.

Operator Documentation

Typical operator documentation would include:

- name of the program or application
- name of the person to be contacted in case of problems
- location of program disks
- date and identification code of latest revision
- description of data inputs required for the program and instructions for obtaining the input data
- instructions for loading and running the program
- distribution instructions including a distribution list and the method(s) of distribution
- frequency of reporting.

Operator documentation internal to the program would include prompts, other instructions displayed on the screen and other explanatory information displayed on the screen during program operation.

The objective of operator documentation is to allow a person unfamiliar with the program to successfully run the application. Conformity of operator documentation to this requirement can be tested by giving the documentation to a person unfamiliar with the program and asking him/her to run the program. An identification code should only be issued to programs whose documentation pass the test.

Data User Documentation

User documentation includes both documentation in the program output and stand alone documentation. The formal documentation must include, at a minimum, a clear statement of the purpose of the program, the assumptions made, limitations of the program, the inputs and outputs of the program, and sample output. The sample output is preferably from a validation test run using historical data that the user can check for consistency with experience. The programmer and user must agree on these matters and both must sign off before the documentation is accepted for controlled programs.

Documentation in the program output should include identification of the version of the program, its date, the program identification code, the date and source of critical input data, operator identification, and a brief description of the output. Warning messages should appear if any input or output is outside predetermined limits.

Specific situations may require expansion of the user documentation described in a general fashion above. No mechanical procedure will catch all problems. The user will always have to use judgment to determine whether the data seem reasonable in the light of experience and current circumstances. It is the user's responsibility to investigate further if something appears to be abnormal. The procedures outlined above will, however, reduce the risk of undetected errors.

Program Controls

Program controls include program elements to avoid or signal possible errors. Some common program controls include:

- use of checksums, footing and crossfooting, and automatic comparison of input and output data with pre-set limits
- protecting all cells of a spreadsheet except those that are to accept data
- use of windows or data entry tables
- use of compiled programs including spreadsheets and data bases
- use of macros to cause automatic recalculation when the operator issues a print command.

This listing of QCP elements is not exhaustive. QCP administrators will add, change and delete elements to create a QCP appropriate to their firm's specific needs. It does, however, provide an indication of the nature of a QCP.

Managerial Vulnerability and the QCP

Managers who rely on data from others are vulnerable in three ways: (1) the data they need may not be available when they need it, (2) the data may be available but erroneous, or the (3) data may be valid and available, but the manager may not understand the data as presented. Common causes of each problem will be presented below. A discussion of how the QCP addresses each problem is presented with the discussion of the problem.

Unavailable Data

Lack of data availability can be caused by hardware problems, software problems and operator problems. Hardware problems are the simplest to solve. Other compatible PC's at the same location can provide short-term backup. Service contracts and the relatively low cost of replacement equipment provide viable solutions for hardware failures.

A damaged disk or missing disk can prevent production of data needed by managers. The archival copy of the program maintained by the QCP administrator can be used to make additional copies if the operating department's disks are lost or damaged.

- Operator Problems

Lack of an operator can effectively prevent production of necessary data. Two techniques can be used to counter this problem: operator backup and operator documentation. Training more than one operator to run a program is generally effective, especially if the backup operator(s) periodically make production runs to keep their level of competence high. Maintaining backup operators is not always practical and does not take care of the situation in which the backup operator is also unavailable. Maintaining backup operators should be strongly encouraged, but cannot be the primary basis of control.

Operator documentation can be used as both a method of control and a means of decreasing dependence on specific operators for critical data. Persons with the ability to create spreadsheet templates, data base programs, etc., are too valuable to use for program operation, which should be basically a data entry operation. The programs should, therefore, be designed for simplicity of operation.

Operator documentation and the operating characteristics of the program should allow a person unfamiliar with the program to assemble the inputs required, generate the required report, and distribute it to the proper persons by following the documentation. As discussed earlier, a QCP identification code is only issued to programs with documentation satisfying this requirement.

Erroneous Data

Erroneous data from PC programs is both common and serious. Some of the more basic causes of erroneous output include program errors, data entry errors and use of the wrong version of a program. Errors will be made in writing a program of any significant length or complexity. Inadequate validation testing can allow these errors to remain in the final version of a program.

The QCP policy requires validation tests for critical applications, with the test results to be submitted as a condition for issuing an identification code. This will help in two ways. It will make those writing program more familiar with validation testing and it will make the programs they write much more reliable.

the particular tests should be selected by knowledgeable personnel to be appropriate for both the firm and the type of application under consideration. The nature and scale of testing must have a favorable cost/benefit ration and at times may be substantially less extensive than testing of mainframe programs.

In some cases, PC validation testing might be limited to checking to see that row and column totals match, testing with artificial data such as all ones or 100's that make errors easy to spot and testing with historic data for which the "right" answer is already known. Commercial programs designed to audit specific software packages for specific common problems such as logic errors and circular references are economical and can be very useful. Other applications might require much more extensive testing.

The application developer may need to consult with the QCP administrator, internal audit or EDP personnel, or the firm's CPA's to determine the appropriate testing procedure for especially critical or difficult applications.

- Protecting Against Data Errors

Erroneous output resulting from bad input can occur because of data entry operator error, errors in the source data, and use of the wrong data set for input. These errors can be significant, and the simple techniques that are practical to use in many applications will help, but will not catch all errors.

Use of more than one data entry operator and comparing their input will catch many mechanical errors. This procedure is not always practical, however. Checksums and footing and crossfooting are useful and can be implemented without undue difficulty. Comparison of input data with pre-set limits will catch some data source and data entry errors. Program input should include the source and date of input data. Including this information in the program's output allows both the operator and the user to verify that the correct data source is being used.

Protecting all cells in a spreadsheet except those that are to accept data input can help avoid unintentional modification of the program which could give erroneous output. Use of data entry tables minimizes the input strokes needed to enter data into the program as well as the likelihood of input errors or omitted input. Compiling spread sheets, data base programs and other programs can avoid some errors that result from operator interaction with the program.

A common spreadsheet error is for operators to turn off the automatic recomputation feature of spreadsheets while entering data and forget to turn it back on before printing their results. Use of menus for input can be coupled with macros to cause an automatic recomputation when the operator selects the print option from the menu.

These and similar program features are included in the QCP's checklist of required program characteristics. There are too many techniques to allow discussion of all of them here. The above techniques will cover a substantial portion of data errors, however. The QCP administrator can determine the methods most appropriate to individual firm and department circumstances.

Misunderstood Data

Misunderstood data can be as harmful to the firm (and to the manager's career) as inaccurate or missing data. Misunderstood reports are often the result of the programmer and the user not communicating with each other. The user documentation requirements discussed in the QCP can help avoid this problem.

V. Summary

Many PC programs generate information used in making decisions that can have a material economic impact on the firm. Their economic impact dictates that there be backup, validation and documentation controls applied to these programs to safeguard the assets of the firm. This paper proposes that critical PC applications be brought into the formal information system of the firm by requiring that program documentation and evidence of validation testing for critical programs be evaluated by the AIS function. AIS would maintain backup software and documentation as well as issue identification codes to those programs meeting company standards.

Use of the identification code procedure allows managers to know whether the data they are using have met firm standards in its production. They have the freedom to use any data source necessary, regardless of potential reliability problems. This preserves the flexibility, timeliness, and economic characteristics that make PC generated reports popular while putting the user on notice of risks involved when the programs creating those reports have not been fully evaluated. The manager also retains control over his/her programmer's output. These factors should make the control package more acceptable to managers.

Several specific techniques that can be applied to spreadsheets, data bases, etc. are presented. Firms that already validate and document their critical PC programs will find the cost of providing that documentation to the QCA administrator is minimal. Firms that do not do this already will face some costs, but will reduce the risk of a disaster by adopting these procedures.

William Paxton, *D.B.A, CPA is an Assistant Professor of Accounting at Cleveland State University, Cleveland, Ohio. He is a member of the Ohio Society of CPAs, the AICPA, the AAA and the NAA.*

Creeth, Richard. "Microcomputer Spreadsheets: Their Uses and Abuses," Journal of Accountancy, June, 1985, pp. 90-93.

Greenberg, Ken. Launching Javelin, PC World March 1986, pp. 144-149.

Howitt, Doran. "Avoiding Bottom Line Disaster," InfoWorld February 11, 1985, pp. 26-30.

Merino, Barbara D. "Personal Computers: Asset or Liability," Journal of Accountancy, April 1983, p. 106.

Kseniak, Mark. "How Personal Computers Can Trip Up Executives," Business Week, September 24, 1984, pp. 94, 98, 102.

PC Week. "The Biggest Blunder: Flawed Spreadsheets Are Sought for Contest," PC Week July 27, 1986, p. 34.

Powell, Norma C. and Strickland, Sherre G. "Security in the Microcomputer Environment," The Ohio CPA Journal, Autumn 1989, pp. 20-23.